TRABAJO FIN DE GRADO

ESTUDIO DE RENDIMIENTO DE ASISTENTES VIRTUALES DE VOZ EN CONDICIONES RUIDOSAS

GRADO EN INGENIERÍA DE TECNOLOGÍAS Y SERVICIOS DE TELECOMUNICACIÓN

ESTUDIANTE: JAVIER LÓPEZ LÓPEZ

TUTOR: JOAQUÍN GONZÁLEZ RODRÍGUEZ

FECHA: JUNIO, 2020

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

Durante la última década, el ruido y las voces de fondo en los diferentes dispositivos de asistentes virtuales de voz han sido un reto para grandes empresas como Google, Amazon, Microsoft, Apple, etc.

En el presente trabajo estudiaremos el rendimiento sobre los reconocedores de voz de Google y Microsoft, realizando una labor comparativa entre estos.

Empezaremos realizando una tarea de investigación sobre los asistentes virtuales de voz: conceptos generales, diferentes tipos de asistentes virtuales con sus diferencias en software, etc.

Para finalizar la parte teórica, realizaremos un trabajo de análisis sobre el proceso de desarrollo de una skill o action para Amazon y Google.

Después de realizar este análisis sobre las ventajas e inconvenientes de cada uno de estos, realizaremos una pequeña labor experimental desarrollando una skill para Amazon y Action para Google para poder realizar un estudio más objetivo sobre la información obtenida en la sección anterior.

En la parte experimental de este proyecto, procederemos a analizar y comparar el rendimiento de los reconocedores de Google y Microsoft para comprobar el funcionamiento de los asistentes virtuales de voz en condiciones ruidosas. Google home y Azure serán analizados respectivamente, usando ficheros de audio que se encuentren en estas condiciones.

Para esto, disponemos de una base de datos de ficheros de audios grabados en una cabina de avión, y realizaremos las respectivas transcripciones. Después, compararemos las transcripciones obtenidas con las reales, para poder observar el rendimiento de ambos.

Para finalizar el documento, realizaremos una última comparación entre el rendimiento de los reconocedores en las transcripciones obtenidas y el rendimiento de las transcripciones en el proyecto HiWire, proyecto del cual hemos podido obtener esa base de datos.

ABSTRACT

During the last decade, the noise and the background voices in the different virtual assistant devices have been a challenge for a wide number of companies like Google, Amazon, Microsoft, Apple, etc.

In the present research, the performance of the voice recognizers of Google and Microsoft devices are being monitored and studied in a comparative study.

We are starting with a generic research on the virtual assistant devices such as general concepts, different types, and software differences, etc.

Concluding the theoretical part, the process of developing a skill or action for Amazon and Google is being analyzed thoroughly.

After listing the advantages and disadvantages of each one of the above, we are running a short experimental task, developing a skill for Amazon and an action for Google respectively. That way we are able to gain a more objective insight on both assistant devices.

In the experimental part of this research, the performance of two different voice recognizers, Google, and Microsoft, is compared to solve the virtual assistant software's challenges in noise conditions. Google Home and Azure are analyzed, respectively, using audio files in noise conditions.

In order to do so, we are accessing a database of pre-recorded files taken in a plane cabin and we are transcribing them accordingly. Later, we are comparing those to the real transcriptions, to monitor and observe the performance and accuracy of both recognisers.

Finally, we are conducting a last comparison between the performance of those recognizers and the performance of the HiWire transcription project, from which we acquired the database.

1. INTRODUCCIÓN

1.1. CONCEPTOS GENERALES SOBRE LOS ASISTENTES VIRTUALES Y BOTS

La existencia de los bots se remonta 50 años atrás, cuando el matemático Alan Turing, uno de los padres de la ciencia de la computación y precursor de la informática moderna, tal y como se describe en el documento de su Wikipedia(1), inventó los conceptos que hoy en día hacen funcionar la inteligencia artificial (IA).

El software de los bots conversaciones, también conocidos como chatbots, y el de los asistentes virtuales, se basan primordialmente en dos tecnologías: la **inteligencia artificial** y el **procesamiento del lenguaje**. Gracias a estas tecnologías, hoy en día podemos mantener una conversación con los asistentes virtuales, ya sea con el fin adquirir nuevos conocimientos o para ejecutar acciones sin la intervención del ser humano, como, por ejemplo: Reservar en un restaurante, marcar alguna fecha en el calendario, realizar transferencias bancarias, etc.

- Un asistente virtual es un agente de software que ayuda a usuarios de sistemas computacionales, automatizando y realizando tareas con la mínima interacción hombre-máquina, usando como medio de comunicación la voz.(2)

- Un chatbot, también conocidos como bot conversacional, es un programa que simula mantener una conversación con una persona al proveer respuestas automáticas a entradas hechas por el usuario, siendo estos muy útiles en aplicaciones de mensajería. La gran ventaja de los chatbots es que no tiene la necesidad de ser descargados ni para ejecutarse ni para actualizarse, por lo que, en consecuencia, no ocupan memoria en los dispositivos.(3)

En este documento, estudiaremos las ventajas y desventajas de desarrollar un servicio para dicho chatbot o asistente virtual, o bien contratar los servicios de un proveedor externo. Posteriormente, procederemos a realizar un desarrollo experimental, dónde obtendremos una visión más objetiva sobre los asistentes virtuales de voz.

Antes de meternos en materia, veremos que asistentes virtuales comerciales son los más utilizados hoy en día, realizando diferentes tipos de estimaciones según diversos estudios que mencionaremos a continuación.

Un estudio realizado por IPMARK, portal de marketing online, en abril de 2019(4), estimó que alrededor de 4,3 millones de familias utilizan asistentes virtuales comerciales día a día, lo que equivale al 10,7% de la población mundial, según los resultados obtenidos en ”*La 1º Ola del Estudio General de Medio 2019*” (EGM) y destinados a la *Asociación para la Investigación de Medios de Comunicación (AIMC).*

Contrastando esta información con la que nos ofrece *es.statista.com*, portal de estadísticas oficiales en línea alemán(5), podemos observar una estimación sobre el número de usuarios que usarán asistentes virtuales en sus viviendas.

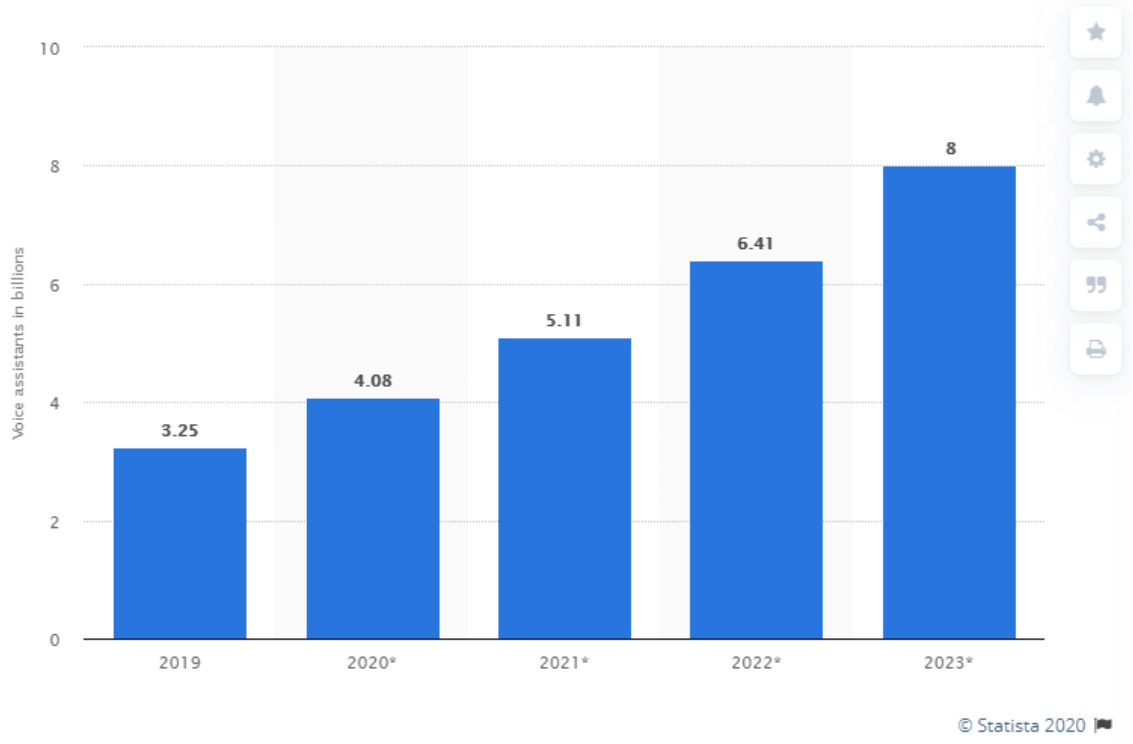

Figura 1

Estimación sobre el número de usuarios que usarán asistentes virtuales en sus viviendas por es.statista.com

Según el informe publicado por la empresa de investigación Clutch en 2019 [(6)], con sede en Washington, podemos conocer los asistentes virtuales más usados. **Amazon Echo**, conocido como **Alexa**, es usado por dos de cada tres personas (66%) que disponen de un asistente virtual, mientras que **Google Home** le sigue con un 40%, mientras que el asistente virtual **Apple HomePod**, solo obtiene un 2%.

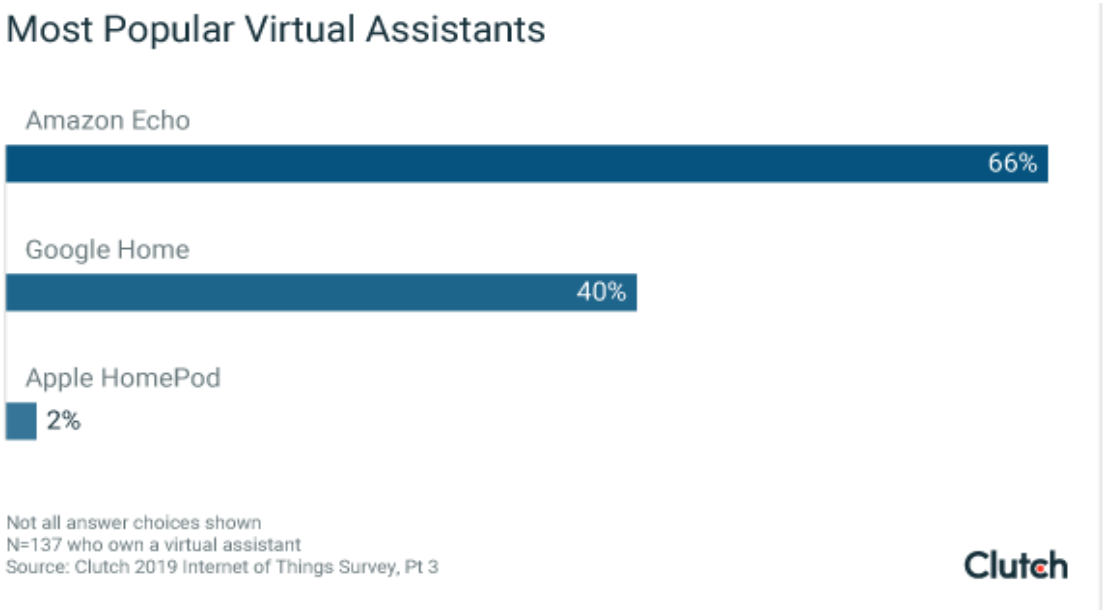

Figura 2

Asistentes virtuales más utilizados según la empresa de investigación Clutch en 2019.

La principal diferencia entre los asistentes digitales como Siri (Apple) o Cortana (Microsoft), frente a los asistentes de voz externos, es que estos últimos son dispositivos independientes de uso principalmente doméstico, como lo son Alexa y Google Home.

La función de voz, mercado que abrió Amazon y fue seguido por Google, hace que esta tecnología sea más humana, la inteligencia Artificial (IA) la hace más inteligente y la conexión con Smart Home mucho más cómoda.

1.2. COMPARATIVA: GOOGLE HOME CONTRA ALEXA (Amazon echo)

Proseguimos documentando las principales diferencias tecnológicas entre los dos asistentes virtuales de voz predominantes que existen actualmente en el mercado.

Comenzaremos con una pequeña explicación sobre las redes neuronales y la inteligencia artificial.

Las **redes neuronales artificiales** consisten en un conjunto de unidades conocidas como neuronas artificiales, conectadas entre sí para trasmitirse señales. La información que atraviesa la entrada recorre la red neuronal realizando diferentes operaciones, hasta obtener unos valores de salida.

Según el documento de la Wikipedia[7], estos sistemas aprenden a la vez que se forman a sí mismos.

Para realizar este **aprendizaje automático**, normalmente, se intenta minimizar una función de pérdida que evalúa la red en su totalidad, hasta conseguir reducir el valor de esta función de perdida. [8]

El **procesamiento natural del lenguaje (NPL)**, extrae información valiosa de textos sin estructurar gracias al aprendizaje automático.

Posteriormente, desarrollaremos el funcionamiento del aprendizaje automático y procesamiento natural del lenguaje (NPL) para ambos asistentes virtuales de voz:

- Google Home: Según el documento redactado por desarrolladores de Google en developers.google.com/machine-learning[9] y en cloud.google.com/natural-language [10], Google Home utiliza la **propagación inversa** como algoritmo de entrenamiento de sus redes neurales.
  Esto hace que el **descenso por gradiente** (algoritmo que minimiza la función de pérdidas), sea posible para redes neurales de varias capas.
  El algoritmo se desarrolla en la plataforma TensorFlow, plataforma de código abierto para el aprendizaje automático, ya que esta herramienta realiza la propagación inversa automáticamente.

  NPL de Google, utiliza dos herramientas: AutoML Natural Lenguage y API Natural Lenguaje:

  - AutoML Natural Lenguage: Interfaz de usuario que te ofrece la posibilidad de crear tus propios modelos de aprendizaje automático personalizados y de alta calidad.
    Así podemos extraer, preparar, y probar tu propio modelo a partir de artículos, pdf escaneados u otros archivos.

  - API Natural Lenguage: Modelos preparados previamente por los desarrolladores de Google.
    Trabaja con funciones de comprensión del lenguaje natural, análisis de opinión, análisis de entidades, clasificación de contenido y el análisis sintáctico.

    En la siguiente imagen, vemos cómo la API Natural lenguaje de Google analiza los textos clasificando cada tipo de palabra:

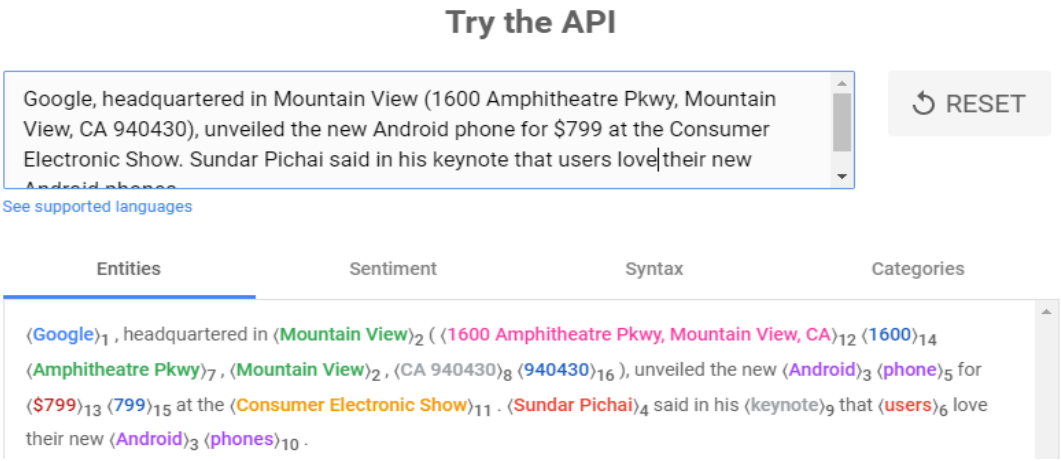

Figura 3

Demostración de la API Natural Lenguage en la plataforma cloud.google.com/natural-lenguage?hl=es

- Alexa: Tal y como se muestra en el informe de Forbes.com, **Alexa Voice Service (AVS)** es el encargado de interpretar la voz que recibe Alexa.

  Junto a este servicio, **Natural lenguage understanding (NLU)** proporciona a los dispositivos el contexto necesario que hay detrás de lo que decimos, y la flexibilidad para comprender las variaciones que existen en cómo podemos decir cosas idénticas.

  Esta técnica es muy similar a la figura 2.1 de Google Home [(11)]

  El NPL de Amazon, en utilizado **en redes neuronales profundas (DPP)** para el procesamiento natural del lenguaje.[(12)]

  Principalmente se utiliza el algoritmo GMM descrito a continuación:

  - GMM: Modelo de mezcla gaussiana, es un algoritmo de agrupamiento dónde cada grupo se modela de acuerdo con una distribución gaussiana diferente.

1.3. Otros Asistentes Virtuales

Siri de Apple:

Según la última revelación de Apple, en la conferencia de enero de 2018 en la que tuvo lugar el lanzamiento del dispositivo HomePod, anunció que Siri estaba ya disponible en 500 millones de dispositivos, lo que implicó en ese momento, haber superado a su competidor; El asistente de Google.

[13]El asistente virtual de Siri consta de un software privado, cuya última versión es iOs 13, como se indica en su propio sitio web.

Las limitaciones de Siri a la hora de desarrollar diferentes tipos de skills son muchas, pero según el portal de publicaciones mundiales referenciadas *Medium*, Apple tuvo un punto de inflexión en los diferentes softwares iOs:

- Con el lanzamiento de la versión de software iOs 10, Apple lanzó Sirikit.
- Con el lanzamiento de la versión de software iOs 12, Apple lanzó Siri shortcut, lo que proporcionó a los desarrolladores las herramientas necesarias para crear su propia interacción personalizada configurable por parámetros.

A pesar de estos esfuerzos realizados por Apple, Siri sigue teniendo muchas limitaciones, ya que Siri shortcut solamente te permite hacer desarrollos cómo crear interacciones o servicios directos a aplicaciones, pero no permite personalizar el código por medio de un usuario.

A continuación, podemos observar algunas de los mayores inconvenientes de Apple a la hora de desarrollar para su plataforma:

- Apple requiere que los usuarios ya tengan una aplicación en una de las plataformas de Apple para poder desarrollar y publicar cualquier skill.
- Esta habilidad también tiene que diseñarse con el lenguaje predeterminado de Apple debido a su software privado, mientras que los demás asistentes virtuales te permiten desarrollar en un lenguaje conocido por gran parte de los desarrolladores; JavaScript
- No permite la personalización del código.

En mi opinión, a menos que se quiera expandir la aplicación que ya tenemos creado en otro dispositivo Apple, podemos omitir Apple hasta que permitan un desarrollo más independiente para su dispositivo.

Bixby de Samsung, Cortana de Microsoft:

Bibxy, el asistente virtual disponible en dispositivos Samsung, junto con Cortana, asistente virtual de Microsoft, representan una minoría en comparación con los asistentes virtuales de Google, Amazon, y Apple.

Centrándonos en el informe redactado en la página web inglesa de noticias y artículos, uk.reuters.com[14], Amazon y Microsoft anunciaron que sus respectivos asistentes virtuales podrían funcionar juntos a partir de 2018, y es gracias a esta actualización, dónde por primera vez dos asistentes virtuales pueden trabajar conjuntamente con la información recopilada por ambos.

2. DESARROLLO DE SKILLS DE ALEXA Y ACTIONS DE GOOGLE HOME

2.1. Proceso del desarrollo de Skills (Amazon) o Actions (Google)

Según el artículo publicado por Laura Reynaud[15], Pre-sales Engineer en Ibenta, nos explica los pasos a seguir para desarrollar una skill o action, y como saber si podemos desarrollar esta skill por completo, o bien utilizar la ayuda de un proveedor externo.

Desarrollar una skill o action, es lo mismo que construir un fragmento de código que utiliza un conjunto de datos (base de conocimiento), con una interfaz de usuario.

1. Esta skill puede incorporase directamente a la consola de Amazon o de Google Home.
2. La importancia del training de esta skill será el matching que habría que enseñar al altavoz, de entre todas las preguntas potencialmente formuladas, obtener una respuesta gracias a la unidad de conocimiento, tarea que consume mucho tiempo y recursos.

   Sin embargo, si ya trabajamos con una solución de procesamiento de lenguaje natural (NLP), tenemos la ventaja de poder obviar el proceso de training vinculando directamente mi aplicación de altavoz inteligente con mi tecnología NLP, gracias a las API's.

Pongamos un ejemplo práctico del caso de Laura Reynaud redactado en su artículo:

- Quiero desarrollar mi skill o action, pero en vez de compartir ese código, conocimiento que nos pertenece, mi propia aplicación envía cualquier pregunta a la interfaz de usuario (a Alexa o Google Home), a la empresa (en este caso la empresa de Laura, Ibenta), a través de la api de chatbots.

  La propia compañía es la que se encarga de procesar y comprender la entrada (habilidades de NLU y NLP), y obtiene estos matching semánticos a partir de una base de conocimientos.

  Así mis conocimientos se encuentran en esa compañía en vez de en la consola de Amazon o Google.

  La compañía de inteligencia artificial (IA) es la que se encarga de devolver una respuesta correcta a mi aplicación, y el altavoz inteligente, como paso final, se lo comunicará al cliente.

  Conclusión:

  La empresa nos ahorra el training del matching. Quien proporciona la interfaz de usuario, tiene el control de esta. Esto quiere decir, si quiero implementar mi skill en Alexa, por ejemplo, debo solicitar a Amazon que lo valide, y Amazon tiene la última

palabra para decidir si mi skill cumple con sus requisitos, tanto para la entrada que recibo como la respuesta que da.

Para diálogos complejos, podríamos crear árboles de decisión, guiones paso a paso a través de mi base de conocimiento, para así ofrecer una conversación humana realista.

Veamos dos imágenes según el documento publicado por Tris Tolliday en diciembre de 2019 [16], para saber cómo funciona Alexa y Google Home:

Alexa

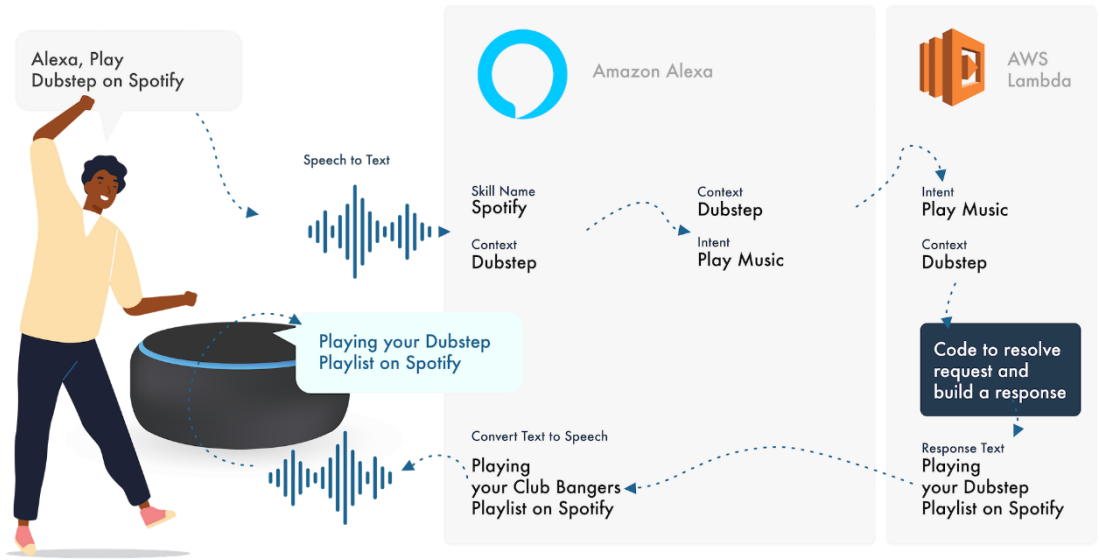

Figura 4

Construyendo skills en Amazon Alexa, publicado en el artículo de Tris Tolliday

Amazon Ecosystem para voz, ha evolucionado para permitir poder desarrollar todas las skills en las propia consola de Alexa, disponible en la web de Amazon.developer.

En un primer lugar, Alexa se ocupa del procesamiento del lenguaje natural (NLP) para poder encontrar una interacción apropiada, la cual va a ser enviada a **Amazon Web Service (AWS) Lambda** para poder tratar esta lógica.

Este servicio devolverá una porción de código a Alexa, que se encargará de convertir estos bits en audio y de forma visual en caso de que fuese necesario, para mostrar en el dispositivo.

Google Home

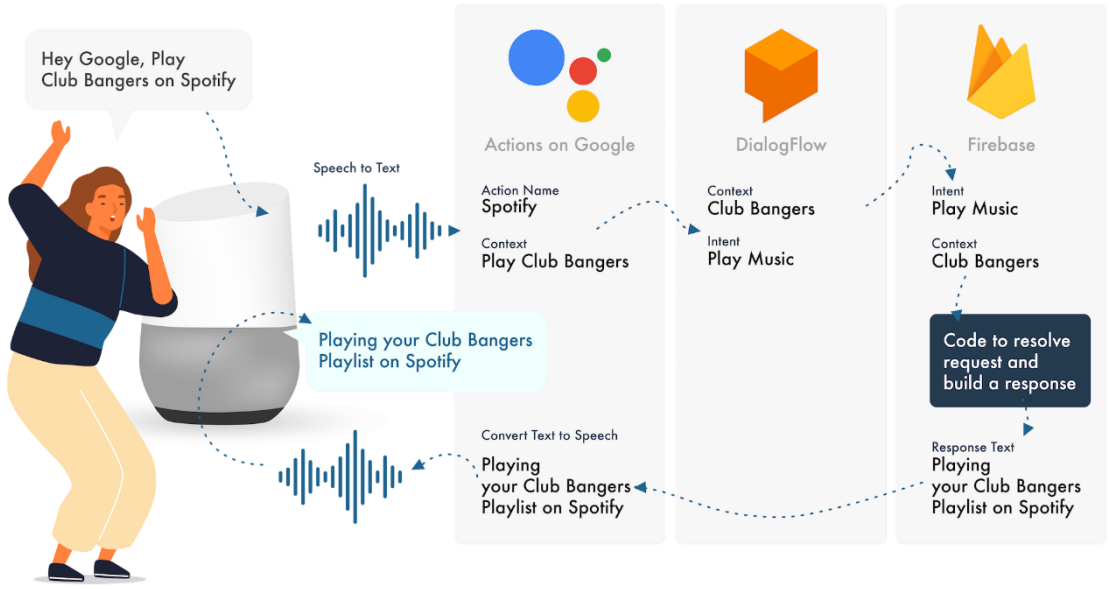

Figura 5

Construyendo skills en Google Home, publicado en el artículo de Tris Tolliday

Al igual que en el proceso de desarrollo de skills para Alexa, un action en Google puede ser desarrollado gracias a la **consola AoG (Actions on Google)** en combinación con **Dialogflow**, ya que te permite mejorar tus habilidades con **FireBase**.

Existen tres partes principales a la hora de desarrollar un Action:

- AoG, se ocupa del PNL (Programación neurolingüística).
- Dialogflow, resuelve las interacciones del usuario.
- FireBase, cumple con la solicitud y envía una respuesta a AoG.

2.2. Programación y diseño de Skills para empresas con Amazon Alexa

Tal y como se muestra en la página web de https://developer.amazon.com/es/ [17], para crear Skills se utiliza ASK (Alexa Skills Kit), conjunto de herramientas, documentación, muestras de código y API en self-service con el que puedes añadir nuevas Skills a Alexa.

Con este kit, puedes aprovechar el conocimiento y la innovación de Amazon en el sector del diseño de voz.

También, te permite complementar tu skill con experiencias visuales y táctiles, para así llegar a nuevos consumidores de forma rápida. Es muy útil para impulsar negocios o nuevos productos, para poder ofrecer publicidad a un gran número de usuarios.

Otra alternativa sería utilizar un proveedor externo como son empresas como: AirTouch, Viiz, Everis, etc.

Adicionalmente, puedes presentar la mayoría de tus skills con créditos promocionales de AWS, y tener la opción de ganar dinero con estas skills, con una variedad de ventajas como: Programas de beneficio donde tienes la opción de ganar créditos, certificación de desarrollador de skills para Alexa (título muy valorado a nivel profesional), etc.

AWS tiene también servicios gratuitos para desarrollar skills, que cuenta con más de 1 millón de solicitudes de AWS Lambda, y 750 horas de tiempo de cómputo de Amazon EC2 por mes sin encargo.

2.2.1.Caso de éxito con Alexa

Tal y como dice el artículo disponible en los blogs de developer.amazon.com [(18)], Steven Arkonovich, profesor de filosofía y ética, nos cuenta su experiencia como desarrollador, ya que ha diseñado 1 de las 7 skills más imprescindibles de Alexa. Actualmente tiene más de 60.000 usuarios mensuales.

- Propuesta:

La Skill se llama “big Sky”, servicio gratuito, pero cuando las compras en habilidades se hicieron disponibles, Arkonovich pudo ver la oportunidad de mostrar un contenido personalizado premium a los clientes, y poder monetizar la suscripción de los clientes.

Dicha habilidad ofrece información meteorológica personalizada con la cantidad de detalle que el cliente solicita, y así poder desbloquear funciones premium para permitir a los clientes personalizar más su experiencia (tiempo, unidad de medidas, porcentaje de precipitaciones, velocidad del viento, etc.).

- Solución y repercusión:

Gracias a estas compras en habilidades, las suscripciones semanales de Arkonovich aumentaron un 486%, pudiendo monetizar el dinero de cada suscripción. Tal y como dice Arkonovich: “Ahora me veo convirtiendo mi pasión por desarrollar las habilidades de Alexa en un negocio real".

Para finalizar, si queremos expandir nuestro negocio o nuestro área de clientes, debemos tener en cuenta que Alexa es el asistente virtual de voz más utilizado actualmente, como hemos descrito al inicio de este documento.

2.3. Programación y diseño de Actions para empresas con Google Home

Tal y como describe la propia plataforma de https://developers.google.com/ [(19)], Actions on Google es una plataforma que permite establecer una integración continua de tus servicios

en el asistente de Google y así poder llegar a más de mil millones de dispositivos, gracias a la cantidad de dispositivos Android que tienen el asistente de Google incorporado.

Google permite utilizar su SDK para poder utilizar sus herramientas disponibles para crear estos servicios, con su propio equipo de asistencia y de revisión técnica que es el que tendrá la última palabra en aceptar o no la action creada, según la política de la empresa.

Similitudes con Amazon Echo:

- Google te permite desarrollar Actions de manera gratuita, según si el tipo de desarrollador va a ser un usuario o un equipo de una empresa.
- Existen paquetes de Dialogflow, con costes que varían según la cantidad de solicitudes y la duración total del audio procesado, al igual que en Amazon Echo.
- El paquete gratuito tiene un mayor número de limitaciones en el número de solicitudes por minuto que se pueden realizar.
- Puede probar tu Action con el servicio disponible: *Speech to text*, desarrollando la action con interfaces muy ligeras e intuitivas.
- Podemos recibir una compensación monetaria a pesar de que seamos un usuario aficionado.

Diferencias con Amazon Echo:

- Google Home también te permite crear la interfaz que tendrá asistente conversacional, para así poder dotarle de la impresión que nosotros queramos dar.
- Google Home te da la oportunidad de desarrollar una Action sin líneas de código, simplemente utilizando distintos simuladores web de Google Home.
- Google Home te permite probar tu action con tu propio micrófono, a diferencia de Alexa, donde tienes que escribir la entrada que le dirías al asistente virtual de voz.

Propósito comercial de Google:

El propósito de Google ha sido la monetización de Google Asistant a través de anuncios, ya que, como buen servicio de la sociedad de la información, Google está más interesado a medida que gana dinero contigo.

Para conseguir esto, sería necesario que la publicidad que se ofrezca tenga valor, y no sean simples anuncios como ocurre cuando abrimos un vídeo en YouTube, ya que se puede emitir publicidad sin llegar a ser un estorbo para el usuario, y es en lo que Google se centra.

Como explicó la directora de administración de productos de Google, Jennifer Liu, la diferencia que hay entre el buscador de Google tradicional y el asistente de Google es muy simple:

En el buscador tradicional, los especialistas de márketing de diferentes empresas o pymes, pagan por usar la función de anuncios en caso de que se realice una búsqueda relacionada a los productos que venden. Sin embargo, en el caso del Asistente Virtual, este mismo es el que se encarga de extraer los datos de la misma opción que ofrecía el buscador tradicional.

2.3.1.Caso de éxito con Google Home

Tal y como describe la empresa tecnológica y de servicios financieros GFT [(20)], vamos a describir el siguiente caso de éxito: Servicio al cliente de Bankia con Google Assistant.

- El reto:

El principal propósito de Bankia era mantener el ritmo de la evolución de cada cliente y seguir las tendencias y necesidades del mercado. Se estaba buscando nuevas formas y mejores de atener a los clientes utilizando tecnologías modernas, por lo que el departamento de innovación de Bankia optó por utilizar la voz mediante asistentes virtuales para así aumentar los existentes canales de atención al cliente.

Este proyecto buscó evaluar cómo podría el asistente virtual aprovechar al máximo el procesamiento de lenguaje (NPL) para resolver consultas específicas de clientes y casos de uso.

Como Google Asistant se lanzó hace relativamente pronto, en 2016, y a pesar de que la tecnología era muy estable, existía un factor en contra que iba a dificultar mucho como se iba a gestionar el proyecto de Bankia: Google Assistant se actualiza continuamente, lo que dificulta mucho el trabajo.

- Propuesta:

La propuesta de este proyecto consistía en que Google Assistant pudiera detallar los viajes de los clientes, definiciones de nuevas funcionalidades, y casos basados en el comportamiento real del cliente, como, por ejemplo:

- Encontrar sucursales y cajeros automáticos.
- Establecer una sucursal como “mi favorita” y obtener información adicional, como horarios de apertura, número de teléfono…
- Conectarse a la página web de Bankia para obtener información sobre productos financieros específicos.
- Conectarse a la banca móvil para ejecutar transferencias o ver el saldo.
- También es capaz de responder al lenguaje inapropiado.

- Solución:

La solución de Bankia consistió en un ciclo de iteración, que consistía en una mejora continuad, y así se conseguía ofrecer respuestas cada vez más informativas a las preguntas de los clientes.

El banco revisa regularmente las preguntas que los usuarios realizan con más frecuencia al asistente virtual, para agilizar el acceso a dicha información. La información generada por el sistema también es una herramienta de gestión útil.

- Repercusión en los clientes:

Los comentarios de los clientes fueron abrumadoramente positivos, con una puntuación de 4,6 sobre 5, y Bankia es considerado el banco mejor calificado en la Plataforma de Asistente virtual de Google.

2.4. Conclusiones prácticas sobre el desarrollo de Skills y Actions

En esta sección, después de haber realizado una Skill de Alexa y un Action de Google, analizaremos las principales diferencias entre estos para poder realizar una labor comparativa de una manera más objetiva.

En un primer lugar, vamos a estudiar el proceso de desarrollo de una skill sencilla para Alexa con unas simples preguntas y respuestas.

La primera opción interesante que te ofrece Alexa es poder construir tu skill sobre la base de otra, con un modelo preconstruido. También te ofrece la opción de almacenar tus propios recursos back-end de tu skill, o que Alexa los almacene por ti, de esta manera, podremos acceder al editor de código, que te permitirá implementar código directamente en AWS Lambda desde la consola del desarrollador. Lo podemos observar en la siguiente fotografía obtenida del propio sitio web de Amazon.

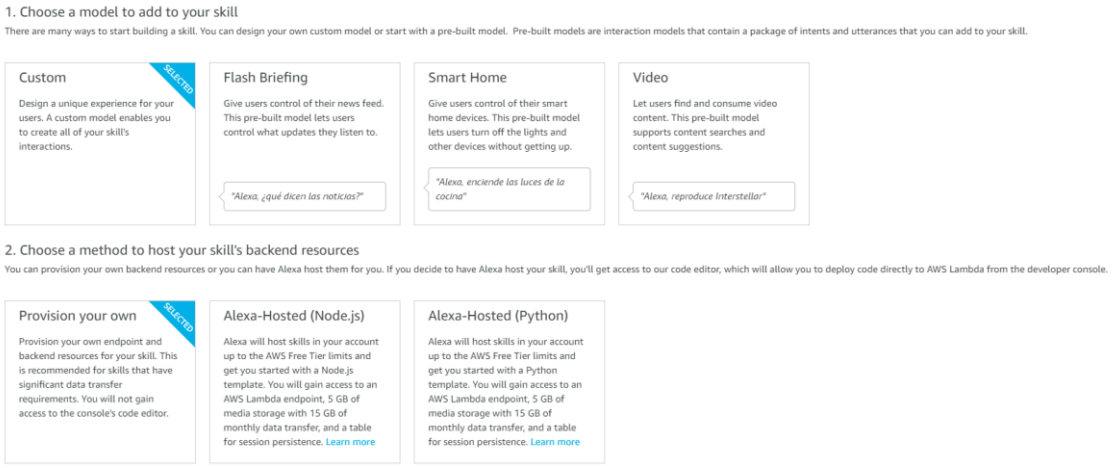

Figura 6

Desarrollar una skill en amazon.developer [21]

Después de acabar de desarrollar una skill, podemos llegar a varias conclusiones.

La manera en la que un usuario puede crear una skill, es un poco enrevesada. Amazon te da algunas pautas muy básicas sobre cómo manejarte por el sitio web, pero es necesario formarme por otros medios para poder realizar una skill. Te da la opción de utilizar diferentes servicios, añadir tus propias líneas de código, navegar y crear árboles de diálogo con tu skill, pero todo son cosas en las que cada usuario tiene que aprender por sí solo.

Todas las habilidades y servicios que queramos añadir tendrán que ser conocimientos que ya tengamos, o necesitaríamos formarnos por cuenta ajena.

Según la página web https://developer.amazon.com/alexa/console/ask, podemos observar estas diferencias en las siguientes pestañas:

- Build →Disponemos de guías básicas y programación visual y sencilla para empezar a desarrollar una skill muy básica.
- Code →Podemos añadir el código de queramos y necesitemos a nuestro antojo.
- Test →Lugar dónde probamos nuestra skill. (Figura 7).

A la hora de probar la Skill, Alexa no se diferencia apenas de Google Home, ya que los dos te dan la oportunidad de probar tu skill en un simulador, dónde la única diferencia que he podido apreciar entre ambos, es que la utilización de tu micrófono para probar de una manera real tus desarrollos, solo está disponible gratuitamente en Google Home.

A continuación, podemos observar dos pruebas de estos desarrollos.

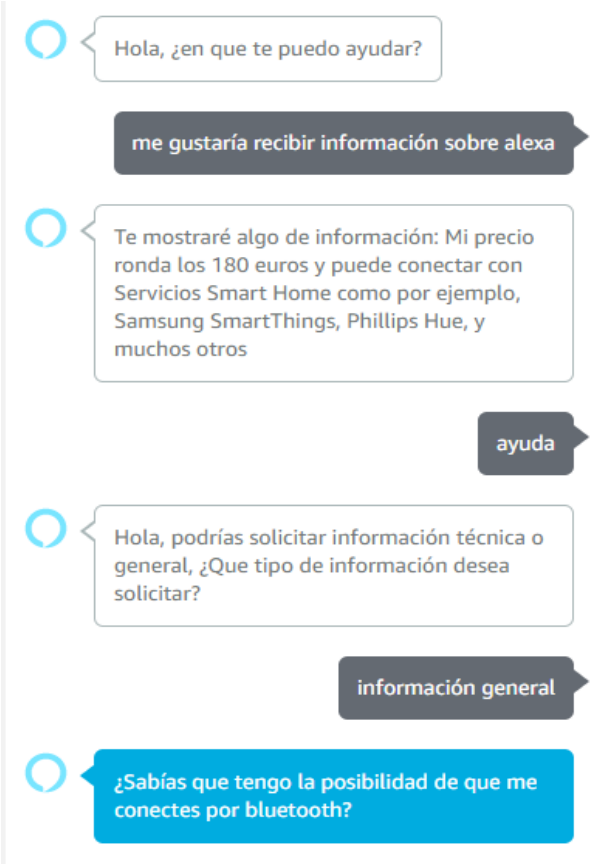

Figura 7

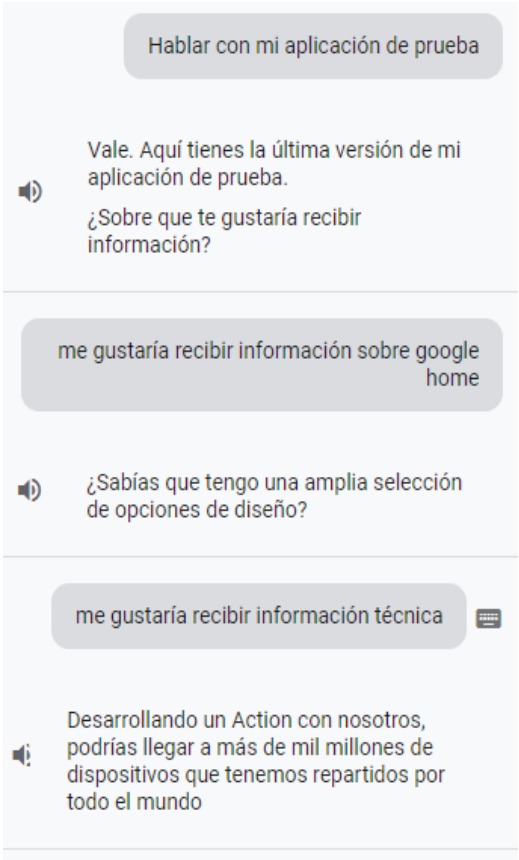

Figura 8

Figura 7: Probando tu skill, https://developer.amazon.com/alexa/console/ask#/skill/create/
Figura 8: Probando tu action, https://console.actions.google.com/u/0/

La conversación que vemos en las figuras 7 y 8, son conversaciones que he desarrollado mediante código e interfaz de ambas plataformas, para poder recibir una respuestas personalizadas posibles preguntas que podría hacer un usuario.

Las pestañas dónde podemos realizar nuestros ajustes en Google, según la web https://dialogflow.cloud.google.com/#/agent, son las siguientes:

- Intents/Entities → Disponemos de guías básicas y programación visual y sencilla para empezar a desarrollar un Action muy básico.
- Fulfillment → Podemos añadir parte de código que necesitamos para un determinado servicio. Actualmente esta función está muy limitada, ya que no podemos añadir todo el código que queramos con distintas funcionalidades.
- Integrations → Lugar dónde probamos nuestra skill. (Figura 8).

Ahora realizaremos un análisis comparativo después de haber realizado un Action muy similar con Dialogflow de Google[22].

La principal diferencia que he podido apreciar a la hora de desarrollar esta Action, son las guías que te ofrece Google para poder ir desarrollando tu Action a medida que vas siguiendo paso a paso la guía, por lo que facilita la manera de desarrollar tu habilidad.

También te ofrece muchos servicios externos, para tener la oportunidad de consumirlos, pero para poder ejecutar esto, sería necesario realizar guías muy avanzadas disponibles en la propia web de Google.

Los puntos clave que diferenciaría con Alexa, sería el mencionado anteriormente, una forma de desarrollarlo mucho más sencillo que Alexa, incluyendo la opción de editar tu agente, poder seleccionar la voz y el género de este, y la oportunidad de probar tu Action con tu propio micrófono.

También existe una desventaja, Google Home no te da la opción de añadir código libre como te proporciona Alexa, primero sería necesario añadir un servicio, y a partir de ahí modificar alguna línea de código en concreto.

En conclusión, Dialogflow te da la oportunidad de aprender de una manera sencilla y útil, mientras que AWS lambda ofrece a los desarrolladores implementar su propio código de una manera más amplia y libre.

Por lo que, en este sentido, los desarrolladores, en mi opinión, tienen más opciones de crecer técnicamente desarrollando Skills para Amazon que Actions para Google.

3. DESARROLLO EXPERIMENTAL

3.1. Introducción

En esta sección se redactará un resumen sobre las tareas a realizar en este desarrollo experimental, y el fin de este:

Nuestro objetivo será poder conocer el **rendimiento del reconocedor de Azure y Google Home**, con el fin de acabar realizando una labor comparativa entre estos.

Para esto, desarrollaremos las siguientes tareas:

1. Análisis de la base de datos que disponemos, más algunas modificaciones de los datos para poder trabajar con ellos.
2. Por cada fichero de audio de esta base de datos, obtendremos su transcripción usando el software de reconocedor de voz de Azure y Google Home.
3. Modificaremos estas transcripciones con el fin de poder trabajar con ellas para una final labor comparativa.
4. Obtendremos el porcentaje de aciertos, errores, frases correctas, etc. por cada fichero transcrito.
5. Obtendremos los datos y procederemos a comparar que reconocedor tiene un mayor rendimiento.
6. Por último, realizaremos otra labor comparativa con respecto al proyecto original del cual hemos obtenido esta base de datos.

El objetivo de esta parte experimental es procesar distintos ficheros de audios mediante el código utilizado por Google Home y Azure para poder obtener la transcripciones de estos.

Después de obtener las respectivas transcripciones, realizaremos un análisis dónde podamos ver el porcentaje de acierto de cada una de ellas.

3.2. Descripción de la base de datos HIWIRE

En la presente sección experimental, vamos a disponer de la base de datos HIWIRE.

Los datos de trabajo disponibles en esta base de datos y de los que disponemos son los siguientes:

- 8000 ficheros de audio que se dividen entre hablantes franceses, griegos, italianos y españoles.
  Hemos analizado 5 hablantes por cada una de estas nacionalidades. Por cada hablante disponemos de información sobre su género, edad, nacionalidad y lengua nativa.
  Por cada hablante tenemos 100 ficheros de audios cortos, entre 1 y 10 segundos, es decir, por cada país de los descritos anteriormente, tenemos 500 ficheros de audio, lo que hacen un total de 2000, pero tenemos réplicas de todos estos ficheros de audio añadiéndole más ruido.
  Para generar los audios con ruido, se mantiene el nivel de voz, y modificaremos la amplitud del ruido para conseguir la relación señal ruido (SNR) deseada.

Podemos encontrar distintos tipos de ficheros de audio:

1. Ficheros limpios (clean).
2. Ficheros con poco ruido (LN, donde la amplitud de ruido se ajusta para obtener valores SNR medios de -5dB).
3. Ficheros con ruido medio (MN, donde la amplitud de ruido se ajusta para obtener valores SNR medios de 5dB).
4. Ficheros con mucho ruido (HN, donde la amplitud de ruido se ajusta para obtener valores SNR medios de 10dB).

La base de datos de la que disponemos ha sido extraída gracias a la base de datos HIWIRE, corpus de habla inglesa ruidosa y no nativa para la comunicación en una cabina de avión.

Según el documento publicado en la biblioteca digital de literatura científica http://citeseerx.ist.psu.edu/ [(23)], esta base de datos fue lanzada por el proyecto IST-EU STREP HIWIRE, con el objetivo de mejorar significativamente la robustez, flexibilidad y naturalidad de la interacción vocal entre humanos y máquinas para aplicaciones aeronáuticas.

Con el fin de mejorar el **reconocimiento automático de voz (ASR)** en entornos aeronáuticos, deben cumplirse dos objetivos: Una mayor robustez frente al entorno, principalmente al ruido, y una tolerancia mejorada al comportamiento del usuario, es decir, tolerancia a diferentes hablantes, acentos, hablas no-nativas, etc.

En la siguiente imagen podemos observar un gráfico en el dominio del tiempo en un entorno limpio (clean), con poco ruido (LN), con ruido medio (MN) y con ruido alto (HN).

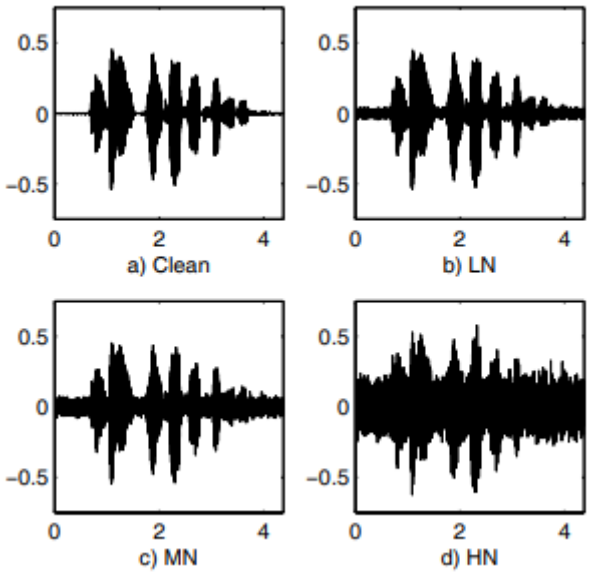

Figura 9

Dominio del tiempo en entornos de ruido limpio (clean), poco ruido (LN), ruido medio (MN), y ruido alto (HN)

La relación señal/ruido (SNR) media de todas las grabaciones originales antes de añadir ruido son de 30 dB.

Información adicional sobre los ficheros de audio:

- Estos audios han sido grabados concretamente en un avión Boeing 375, por lo que nos encontraremos muchos audios con tecnicismos del alfabeto aeronáutico, como por ejemplo F (Foxtrot), B (Bravo), Z (Zulo), W (Whiskey), L (Lima)…

- Han sido grabados con un micrófono sencillo para un hablante a una frecuencia de muestreo de 16KHz a 16 bits, en formato mono (WAV).

3.3. Descripción de los sistemas desarrollados

A continuación, se detalla un análisis exhaustivo sobre la extracción de características de los ficheros de audio y cómo hemos procedido a implementar el código necesario en nuestro ordenador para realizar las trascripciones de cada audio.

La parametrización del habla se basa en las características estándar de MFCC implementadas en HTK. La señan de entrada se divide en ventanas superpuestas de 32 ms de longitud y 10 ms de desplazamiento utilizando una ventana de Hamming. Cada trama se transforma en el dominio de la frecuencia utilizando una FFT de 1024 puntos, y se calculan las energías de la salida de 24 filtros triangulares espaciados por MEL.

La DCT se utiliza para obtener 12 valores MFCC de la salida de la etapa anterior, por lo que el vector de características resultantes contiene 12 MFCC. Este vector de 13 componentes

se aumenta con las características dinámicas de primer y segundo orden que dan como resultado el vector final de características de 39 componentes.

Para mejorar la robustez del vector de características, el CMS se aplica en una oración por frase.

Ahora se procede a describir el proceso de implementación en nuestro ordenador para poder obtener estas transcripciones.

Gracias a la Api de "Speech-to-text" expuestas en la web de Google Cloud y Azure, hemos podido implementar este proceso para poder realizar la parte experimental.

Ambos te dan la oportunidad gratuitamente (hasta un cierto número de audios procesados), de poder utilizar sus librerías como dependencias para poder así llamarlas con un audio nuestro y que lo procese al igual que lo hace para Google Home o Azure.

También tenemos la oportunidad de poder desarrollar esto en distintos lenguajes de programación como son: Java, Node.js, PHP, Python….

Después de configurar nuestro entorno y realizar cambios en el código proporcionado en la web de cada uno de estos asistentes virtuales, hemos procedido a obtener la transcripción de cada audio y guardarla en un fichero txt para cada hablante, es decir, para cada hablante vamos a obtener un fichero .txt con 100 transcripciones de Azure, y un fichero .txt con 100 transcripciones de Google Home.

En la web de cada uno de estos, podemos ver que la transcripción de audios puede ser diferente según la longitud del audio, para tener en cuenta conversaciones largas, que tengan un sentido, en nuestro caso hemos optado por elegir el código de audios cortos ya que como he mencionado antes, son audios entre 1 y 10 segundos.

Después de realizar diversas pruebas, hemos observado que el reconocedor "speech-to-text" de Google se ejecuta más rápido que el de Azure. El reconocedor de Google tarda unos 3 minutos mientras Azure tarda unos 5-6 minutos por cada 100 ficheros de audios cortos.

A continuación, detallaremos las webs de las que podemos obtener esta parte de software de cada reconocedor:

https://cloud.google.com/speech-to-text/docs/libraries#client-libraries-usage-java

https://docs.microsoft.com/es-es/azure/cognitive-services/speech-service/quickstarts/speech-to-text-from-file?tabs=linux%2Cbrowser%2Cwindowsinstall&pivots=programming-language-java

3.4. Análisis de datos y correcciones manuales para las transcripciones generadas por Google y Azure (comparación).

En esta sección veremos de manera analítica las transcripciones obtenidas según el hablante, el idioma, el ruido, etc. Y también sobre los cambios que tenemos que hacer en estos ficheros generados para poder realizar la comparación final con los ficheros de texto que contiene las transcripciones reales.

Problemas encontrados:

1. El mayor problema que hemos encontrado con los ficheros generados es el modo en el que se realiza la transcripción, ya que, si en la transcripción tenemos un número 80, no sabríamos decir si eso es un ochenta o un ocho y un cero, por lo que se tienen que cambiar todos los ficheros a mano, incluyendo símbolos como "-", ".", o hasta frases del alfabeto aeronáutico.
   Por ejemplo, para la frase: "Yankee Victor Delta", la transcripción de Azure nos devuelve "YVD", que es correcto, y la transcripción de Google nos devuelve "Yankee Victor Delta" que también es correcto.

2. Analizando las transcripciones de los audios podemos observar palabras que se confunden muy habitualmente al realizar la transcripción, como, por ejemplo, "two" con "to", "four" con "for", "due to" con "U2", "HF2 three" con "HF 23", "four point zero" con "4.0", etc. Por lo que podemos observar, hay transcripciones las cuales son correctas, pero hay que cambiarlas a mano.

Podemos suponer de antemano que estás transcripciones van a tener un porcentaje de acierto menor al que tienen los asistentes virtuales de Azure y Google Home por dos motivos.

- Las transcripciones hacen referencia a palabras técnicas del alfabeto aeronáutico, y a iniciales que hacen referencia a este lenguaje. (AOC, HF3, VHF3, etc.).

- No hemos definido un reconocedor de gramática para ningún hablante, por lo que el porcentaje de aciertos será menor.

El uso de mayúsculas y minúsculas, no se va a tener en cuenta la hora de comparar los ficheros, ya que el kit de herramientas de puntación SCTK-NIST, no las va a tener en cuenta.

Ya están incluidas en las librería de Google y Azure, palabras como Covid-19, por lo que podemos deducir muy fácilmente que la inteligencia artificial es un trabajo que se actualiza día a día, y en el que nunca va a dejar de existir trabajo.

3.5. Comparación de archivos de texto mediante SCTK.

En esta sección vamos a ver cómo hemos configurado el comparador de archivos de voz en formato .txt, cómo se realiza esta comparación, las restricciones que aplica SCTK, y que datos de salida nos va a mostrar por cada transcripción realizada.

Como se describe en la propia web de esta herramienta, el kit de herramientas de puntuación NIST (SCTK) es una colección de herramientas de software diseñadas para calificar evaluaciones de prueba de referencia de sistemas de reconocimiento automático de voz (ASR).

Antes de empezar a poder descargar este software, es necesario instalarse una máquina virtual de Linux o una terminal de comandos de Linux para poder ejecutar comandos necesarios con el fin de poner en marcha esta herramienta.

Después de descargarnos el software de esta herramienta e instalar las librerías necesarias, solamente tenemos que copiar las transcripciones de Google y Azure, las cuales hemos obtenido anteriormente, y la transcripción real, en una de las carpetas de esta herramienta y ejecutar los siguientes comandos.

```
$ ./sclite -h transcripciones_google.hyp -r list.ref -i wsj
$ ./sclite -h transcripciones_azure.hyp -r list.ref -i wsj
```

Aquí podemos ver que ejecutamos unos comandos llamando a las transcripciones de Azure con la extensión .hyp (hipotética), y llamando también a la transcripción real con la extensión .ref (referencia).

Las restricciones de esta herramienta son las comentadas en el apartado anterior, como, por ejemplo, si reconoce un 20, la herramienta no sabe si es two zero o twenty.

A continuación, describiremos cómo realiza la comparación de ficheros esta herramienta.

Lo primero que va a realiza, es comparar de cada fichero de audio, sus respectivas etiquetas, por lo que, si en cada fichero transcrito hemos dicho que tenemos 100 audios cortos, tendremos 100 etiquetas, una por audio, que se van a reconocer con: (0), (1), (2)…(99).

Lo podemos observar con mejor detalle en la siguiente captura:

```
$ ./sclite -h transcripciones_azure.hyp -r list.ref -i wsj
sclite: 2.10 TK Version 1.3
Begin alignment of Ref File: 'list.ref' and Hyp File:
'transcripciones_azure.hyp'
    Alignment# 1 for speaker 0)
    Alignment# 1 for speaker 1)
    Alignment# 1 for speaker 2)
    Alignment# 1 for speaker 3)
    Alignment# 1 for speaker 4)
    Alignment# 1 for speaker 5)
    Alignment# 1 for speaker 6)
```

Figura 10

Etiquetas Azure de los fichero de audio por la herramienta SCTK para cada hablante, con el fin de obtener el porcentaje de acierto de la transcripción "transcipriocnes_azure.hyp" con respecto a list.ref.

Primero nos muestra la versión que va a utilizar la herramienta, luego empieza a alinear ambos ficheros, el transcrito con el original, y empieza a comparar cada etiqueta 0), 1), 2), etc., para así acabar de alinear ambos ficheros.

Después de alinear ambos ficheros, la herramienta nos va a dar unos resultados que podemos ver y analizar en la siguiente captura:

SYSTEM SUMMARY PERCENTAGES by SPEAKER

transcripciones_azure.hyp								
SPKR	# Snt	# Wrd	Corr	Sub	Del	Ins	Err	S.Err
0)	1	1	100.0	0.0	0.0	0.0	0.0	0.0
1)	1	2	100.0	0.0	0.0	0.0	0.0	0.0

Figura 11

Resultados de la herramienta SCTK por cada transcripción, obteniendo el porcentaje de palabras correctas, sustituidas, borradas, insertadas, erróneas, y frases erróneas

A continuación, describiremos la información que obtenemos de la herramienta SCTK:

- SPKR → Hablante número 1, 2, 3, etc.
- Snt → Sentences, ya que un hablante podría tener dos audios con la misma etiqueta.
- Wrd → Cantidad de palabras totales en las frases por hablante.
- Corr → Correct, nos indica el porcentaje de palabras correctas.
- Sub → Substituted, nos indica el porcentaje de palabras sustituidas.
- Del → Deleted, nos indica el porcentaje de palabras borradas.
- Ins → Inserted, nos indica el porcentaje de palabras insertadas.
- Err → Error, nos indica el porcentaje de palabras erróneas.
- S.Err → Sentences error, nos indica el porcentaje de frases incorrectas.

3.6. Descripción de pruebas y análisis de resultados

Después de realizar todas las pruebas, una prueba por cada 100 audios, hemos podido sacar la media de todos los ficheros juntos, diferenciando por países, por tipos de audios, y por asistente virtual utilizado, etc.

La siguiente tabla recoge solamente las medias finales de cada país y asistente virtual:

AZURE FRANCIA	CORR	SUB	DEL	INS	ERR	SERR
CLEAN	78,9	19,1	1,8	6,9	27,9	39,4
LN	58,5	39,5	1,9	15,8	57,3	59,3
MN	50,6	41,2	8,1	10,3	59,6	66,0
HN	16,6	53,5	29,8	5,0	88,3	91,3

Tabla 1

Resultados de ficheros limpios, con poco ruido, ruido medio y ruido alto, en Azure para hablantes franceses.

Los números nos muestran el porcentaje de aciertos de palabras correctas, sustituidas, eliminadas, insertadas, erróneas, y frases erróneas.

GOOGLE FRANCIA	CORR	SUB	DEL	INS	ERR	SERR
CLEAN	68,8	27,5	3,5	6,4	37,5	54,9
LN	51,0	44,3	44,4	16,7	65,7	71,4
MN	39,1	49,2	11,5	10,1	70,9	81,3
HN	13,2	56,4	10,3	8,0	74,7	75,9

Tabla 2

Resultados de ficheros limpios, con poco ruido, ruido medio y ruido alto, en Google para hablantes franceses

AZURE GRECIA	CORR	SUB	DEL	INS	ERR	SERR
CLEAN	73,2	22,0	4,7	7,7	34,5	40,2
LN	57,8	32,8	9,4	10,7	52,9	57,2
MN	47,5	40,3	12,1	11,7	64,2	67,7
HN	13,8	59,8	26,3	11,6	97,8	93,0

Tabla 3

Resultados de ficheros limpios, con poco ruido, ruido medio y ruido alto, en Azure para hablantes griegos

GOOGLE GRECIA	CORR	SUB	DEL	INS	ERR	SERR
CLEAN	67,8	30,0	2,0	7,8	39,9	52,3
LN	53,1	41,6	5,7	11,3	58,8	68,4
MN	43,4	49,6	6,9	10,1	66,7	76,6
HN	12,5	62,1	23,4	21,9	109,4	96,9

Tabla 4

Resultados de ficheros limpios, con poco ruido, ruido medio y ruido alto, en Google para hablantes griegos

AZURE ITALIA	CORR	SUB	DEL	INS	ERR	SERR
CLEAN	78,5	20,0	1,4	7,0	28,5	33,0
LN	75,2	23,1	1,6	7,4	32,2	37,2
MN	70,4	27,6	1,8	7,6	37,1	42,2

HN	43,0	43,7	13,2	7,1	64,1	67,4

Tabla 5

Resultados de ficheros limpios, con poco ruido, ruido medio y ruido alto, en Azure para hablantes italianos

GOOGLE ITALIA	CORR	SUB	DEL	INS	ERR	SERR
CLEAN	73,7	24,7	1,5	5,2	31,4	44,1
LN	66,0	31,0	2,9	6,2	40,1	52,6
MN	57,4	38,7	3,8	8,7	51,2	60,6
HN	34,7	54,3	10,9	13,8	79,1	81,7

Tabla 6

Resultados de ficheros limpios, con poco ruido, ruido medio y ruido alto, en Google para hablantes italianos

AZURE ESPAÑA	CORR	SUB	DEL	INS	ERR	SERR
CLEAN	72,0	26,1	1,8	9,4	37,4	43,6
LN	67,1	31,1	1,6	10,4	43,1	49,0
MN	62,6	34,5	2,8	9,2	46,6	53,8
HN	28,1	57,4	14,4	12,3	84,2	82,5

Tabla 7

Resultados de ficheros limpios, con poco ruido, ruido medio y ruido alto, en Azure para hablantes españoles

GOOGLE ESPAÑA	CORR	SUB	DEL	INS	ERR	SERR
CLEAN	60,0	36,6	3,3	8,2	48,1	58,8
LN	51,7	44,7	3,5	9,7	57,9	67,4
MN	43,5	51,4	5,1	11,4	67,9	75,4
HN	21,8	62,5	15,6	11,0	89,2	90,2

Tabla 8

Resultados de ficheros limpios, con poco ruido, ruido medio y ruido alto, en Google para hablantes españoles

De estos datos, podemos fijarnos en la parte importante, **palabras correctas** (**corr**), y **frases erróneas (serr)**.

Podemos observar que, para todos los tipos de audios, desde nada ruidosos hasta más ruidosos HN, el porcentaje de acierto para palabras correctas y frases erróneas es mucho mejor para Azure que para Google Home. En la siguiente tabla se puede ver más claramente:

AZURE	CORR	SUB	DEL	INS	ERR	SERR
CLEAN	75,7	21,8	2,4	7,8	32,1	39,1
LN	64,7	31,6	3,7	11,1	46,4	50,7
MN	57,8	36,0	6,2	9,7	52,0	57,5
HN	25,4	53,6	21,0	9,1	83,6	83,6

Tabla 9

Resultados Azure para todos los hablantes

GOOGLE	CORR	SUB	DEL	INS	ERR	SERR
CLEAN	67,6	29,8	2,6	6,9	39,3	52,6
LN	55,5	40,4	14,2	11,0	55,7	65,0
MN	45,9	47,3	6,9	10,1	64,2	73,5
HN	20,6	58,9	15,1	13,7	88,1	86,2

Tabla 10

Resultados Google para todos los hablantes

En la siguiente imagen, podemos apreciar gráficamente el porcentaje de frases erróneas según ambos reconocedores, y llegamos a la conclusión de que el reconocedor de Azure, para este proyecto, tiene un mejor rendimiento que el de Google:

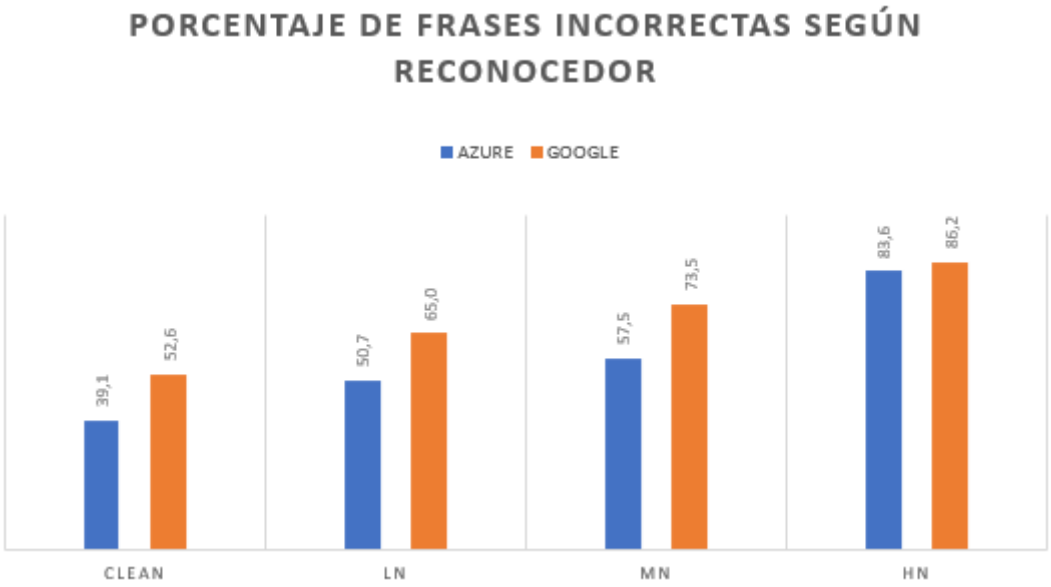

Figura 12

Porcentaje de frases incorrectas según el reconocedor

Aquí podemos ver la media sin distinciones por países. Podemos observar que es algo mejor Azure, como hemos mencionado antes.

Haciendo un análisis exhaustivo sobre los ficheros de cada audio y los resultados, nos hemos encontrado con situaciones en las que los resultados para audios muy ruidosos HN, son mejores que para audios con menos ruidos MN para la misma persona. Después de analizar el audio varias veces, nos podemos encontrar en situaciones en las que, con mucho ruido, la voz suena más clara que con ruido medio, es decir, a veces, con mucho ruido, la voz se escucha de una forma más audible, debido a la diferencia de tonos, y para el mismo caso, con ruido medio, la voz se mezcla con el ruido debido a la frecuencia de ambos sonidos.

Para acabar esta sección, vamos a realizar una comparación con los datos obtenidos en este documento, comparándolos con los datos obtenidos por el proyecto IST-EU STREP HIWIRE comentado al principio de la parte experimental.

A continuación, mostraremos dos tablas, la primera con nuestros resultados globales, y la segunda, con los datos obtenidos en el proyecto HIWIRE.

Resultados globales AZURE + GOOGLE

	CORR	SUB	DEL	INS	ERR	SERR
CLEAN	71,7	25,8	2,5	7,4	35,7	45,8
LN	60,1	36,0	8,9	11,1	51,0	57,9
MN	51,8	41,6	6,6	9,9	58,1	65,5
HN	23,0	56,3	18,0	11,4	85,9	84,9

Tabla 11

Resultado de audios no-nativos por Azure y Google Home

Resultados globales HIWIRE

	CORR	SUB	DEL	INS	ERR	SERR
CLEAN	92,5	6,6	0,8	1,4	8,9	18
LN	45,9	28,9	25,1	0,8	54,9	66,2
MN	23,3	31,8	44,8	0,4	77,1	82,5
HN	2,1	32,5	65,3	0.0	97,8	97,5

Tabla 12

Resultado de audios no-nativos por el proyecto HIWIRE

Vamos a realizar la misma comparativa a nivel de gráfica para observar más claramente los resultados obtenidos. El porcentaje de frases Incorrectas según las últimas dos tablas es el siguiente:

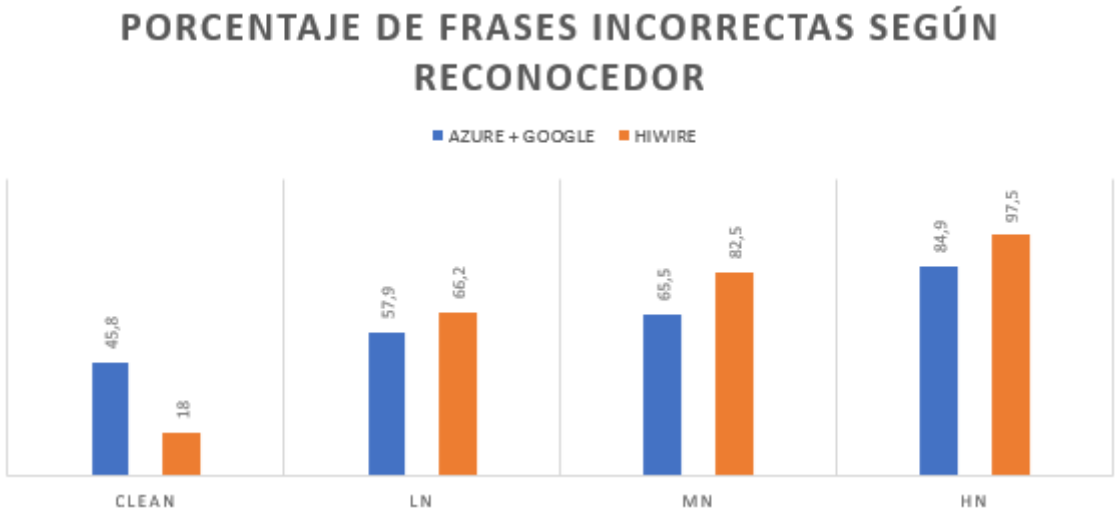

Figura 13

Porcentaje de frases incorrectas según los reconocedores de este proyecto y el proyecto HiWire

3.7. Conclusiones

Después de analizar ambos resultados, podemos llegar a dos conclusiones:

- Como podemos observar, los resultados para audios clean, son muchos mejores los del proyecto HIWIRE que los obtenidos por nosotros, pero estos resultados no son comparables, ya que estamos usando el reconocedor de gramática Hiwire, es decir, solamente contemplamos salidas permitidas por la gramática de Hiwire, reduciendo enormemente la complejidad del reconocedor.

Como los reconocedores de voz que hemos usado en este proyecto han sido Azure y Google Home, es decir, son reconocedores sin una gramática específica para este proyecto, estos tienen que ser capaces de reconocer en cualquier contexto y gramática conocidas, de ahí la diferencia del rendimiento obtenido.

- Para audios en los que se introduce ruido, vemos que los resultados son mejores para Azure y Google que para los resultados del proyecto HiWire.
  Según el catálogo de recursos lingüísticos de ELRA, el proyecto HIWIRE finalizó el 01/01/2007, por lo que, al ser un proyecto tan antiguo, el ruido en los ficheros de audio, hacían del reconocedor una herramienta muy vulnerable.
  Hoy en día, gracias al avance de los reconocedores de audio de Azure, Google Home, Alexa, etc., en entornos ruidosos, ha mejorado bastante, y pueden soportar altos niveles de ruido consiguiendo un funcionamiento aceptable.
  Por eso el porcentaje de aciertos de audios en entornos ruidosos es mejor para el reconocedor de Google y Azure, que para lo que muestra el proyecto HiWire.

4. OPINIONES Y APORTACIONES

Después de realizar el presenta documento, podemos llegar a varias conclusiones:

1. Después de observar que, para nuestra parte experimental, el reconocedor de Google tiene un menor rendimiento que el de Azure, nos ha sorprendido que a pesar de que Google Home es uno de los asistentes virtuales de voz más vendidos hoy en día, Azure tenga un mayor porcentaje de acierto.

2. La labor que hay que realizar para poder empezar a desarrollar Skills y Actions, en mi opinión, es muy sencilla.
   Con simplemente empezar a formarte por tu cuenta, podrías llegar a recibir un título de desarrollador, muy valorado a nivel profesional, de skills para Alexa, y poder hacer de tu hobby, un trabajo.

3. Poder obtener las transcripciones de un audio añadiéndoles ruido u otras voces, hace que nos resulte mucho más sencillo realizar un análisis de ambos reconocedores, ya que estos dos te ofrecen la oportunidad de poder utilizar sus servicios “speech-to-text” y muchos más.

4. Hemos intentado realizar estas mismas transcripciones, pero la web de Amazon: https://us-east-2.console.aws.amazon.com/transcribe/home?region=us-east-2#jobs, la cual ofrece servicios de transcripción, y después de ponernos en contacto con el soporte técnico, no es posible realizar una transcripción automática de varios ficheros de audio simultáneamente.
   Por lo que habría que procesar cada fichero de audio uno por uno y copiar esa transcripción en un fichero de texto.

5. Como hemos mencionado en el resumen, hemos podido analizar el rendimiento de ambos reconocedores para una base de datos dada.

A continuación, se van a detallar las aportaciones realizadas en este documento :

- Estudio y análisis del mercado en referencia a diferentes tipos de asistentes virtuales, con el posterior análisis de software de alguno de estos.

- Tarea de documentación y análisis para la creación de Actions y Skills tanto para Google Home como para Alexa respectivamente.

- Documentación para diversos casos de éxito de ambas empresas.

- Tareas de desarrollo para la creación de una nueva skill y action con el fin de conocer objetivamente ambas plataformas y así poder realizar una labor comparativa entre ambas.

- Para la tarea experimental de este presente documento hemos realizado las siguientes aportaciones:

    - Estudio y análisis de los datos que disponemos, así como de la bases de datos utilizada.
    - Posterior modificación de estos datos para poder trabajar con ellos.
    - Implementación de cada tarea para poder obtener las transcripciones de cada fichero de audio.
    - Configuración del equipo con sus respectivas licencias para poder trabajar con el Git de Microsoft y Google.
    - Instalación del SCTK para poder realizar una labor comparativa entre las transcripciones obtenidas y las teales.
    - Posterior obtención de resultados para conocer la diferencia entre ambos reconocedores de voz.
    - Conclusiones y motivos de las diferencias entre nuestros resultados obtenidos y el proyecto HiWire que trabajó con esta misma base de datos.

5. REFERENCIAS

[1] Alan Turing, Wikipedia

[2] Asistente virtual, Wikipedia

[3] Bot conversacional, Wikipedia

[4] Los asistentes virtuales, utilizados en 4,3 millones de hogares, IPMARK, abril 2019

[5] Previsión del numero de asistentes virtuales en uso a nivel mundial de 2019 a 2023, Statista, febrero 2019

[6] Virtual Assistants and Consumer AI, Grayson Kemper, Clutch, febrero 2019

[7] Red neuronal artificial, Wikipedia

[8] Aprendizaje automático, Fernando Sancho Caparrini, Dpto. Ciencias de la computación e inteligencia artificial

[9] Machine Learning, google.cloud

[10] Natural Language, google cloud

[11] How Amazon Alexa works? Your guide to Natural Language Processing (AI), Alexandre Gonfalonieri, Towardsdatascience, Noviembre 2018

[12] Amazon Comprehend, aws.amazon

[13] Building an Interactive Voice App Using Custom Siri Shortcuts in iOS 12, Alfian Losari, medium, noviembre 2018

[14] Amazon launches initiative to bundle virtual assistants on single device, Techonology news, septiembre 2019

[15] Cómo dotar mi altavoz inteligente de NLP, Laura Reynaud, Zonamovilidad, Noviembre 2019

[16] Creating Voice skills for Google Asisistant and Amazon Alexa, Tris Tolliday, smashingmagazice, diciember 2019

[17] Desarrollo de skills para amazon Alexa, developer.amazon

[18] Steven Arkonovich adds in-Skill Purchasing to Personalize Alexa Skills and Boost his voice Business, Alexa blogs, developers.amazon

[19] Integrate with the Google Assistant, developers.google

[20] Bankia boosts customer service with Google Assistant, Gft

[21] Alexa Developer Console, amazon.developer

[22] Actions on Google, cloud.google

[23] The HIWIRE database, a noisy and non-native English speech corpus for cockpit communication, Citeseerx